Technical Peer review

Reviewing each other’s code (paired assignment)

In this assignment you are asked to review parts of each other’s code on various aspects that have been covered in OOD.

**What to do:**

1. Your tutor will pair your group up into pairs of two.
2. Together with your tutor you decide what code base you will assess as a pair (code that you did not develop yourself).
3. You answer the questions below before the final meeting in week 15.
4. In the final meeting in week 15 you present/discuss your answers with the tutor and the other pair.

|  |  |  |
| --- | --- | --- |
| **Student name 1** | Rositsa Nikolova | |
| **Student name 2** | Daniil Blagoev | |
| **Assessed code base** | Leave\_Preference\_Request | |
| **Date** | 02-Jun-22 | |
| **Does the target code apply inheritance to generalize their code where applicable?** | | No |
| If not, where do you foresee possible cases for inheritance?  Leave\_Preference\_RequestManager and Leave\_Preference\_RequestRepository could implement interfaces | | |
| **Does the target code apply Single responsibility to isolate individual responsibilities?** | | No |
| If not, what classes would you propose that split up (elaborate about this)?  Leave\_Preference\_RequestManager could be split into LeavePreferenceManager and RequestPreferenceManager. Leave\_Preference\_RequestRepository could be split into LeaveRequestRepository and PreferenceRequestRepository. | | |
| **Does the target code apply the Open-closed principle to allow extension of behaviour without modification of existing classes in places where change/extension is expected?** | | Yes |
| If not, where do you expect change/extension to happen, and how would you propose to facilitate this?  Click or tap here to enter text. | | |
| **Does the target code apply the Liskov principle to take benefit of polymorphism?** | | No |
| If not, how can the target code change to communicate in the same way with child objects as you do with parent objects?  Polymorphism is not needed in the target code. | | |
| **When applicable, what other object-oriented design principles are applied in the target base (e.g. interface segregation, dependency inversion, etc.)?** | | |
| Open/Closed Principle | | |

|  |  |
| --- | --- |
| **Is the target code readable (clear naming convention, conscious use of white spaces, proper tab use (indentation)).** | No |
| if not, what could improve?  Methods should begin with a capital letter, white space could be used before and after each method, names are clear, indentation is good. | |
| **Below you have space for any other tips you want to share with the programmer of your target code?** | |
| Everything that could be improved upon was mentioned in the above fields. | |